Introduction to Computer Gaming

Introduction to Computer Gaming – Mod 003217

Max Topan – SID: 1405715

Presentation on 3 March 2015

# Intro

Project Description

This project has been to create the demo for a game within the space of roughly 3 months. This involved meeting the criteria requested by clients wanting to build a game in the Unity 3D game engine

Game Outline

‘Bears: An Adventure in Two Halves’ is a top-down tile puzzle game.  
An inventor creates a robot to move whenever he moves, and realises he needs a wide open space to test it. He enters the woods with the robot and boots it up, but before he can get into the tests he is ambushed by a bear! He realises he has accidentally shut the robot into a warehouse.  
He must now navigate through the dangerous woods to get to the other end of the warehouse, and reunite safely with his robot.

The player controls the inventor and the robot in tandem, creating unique puzzle solving opportunities involving unlocking doors and being able to navigate to the exit, whilst also focusing on both character environments at once.

There is also a rank rewarded to the player at the end of each level based on how many steps they took to complete the level. Gold is the best rank, given for the optimal route in each level.

Included in the demo is a level editor, allowing the player to create and save their own custom levels. This is important for the end-game, when the player has exhausted all pre-made content. Players can also download or create additional level files

Work Methodology

Throughout the development process, the work methodology for Bears has been combined from multiple different methodologies outside of, and within, the games industry.

Before a substantial period of work roughly eight CDs would be chosen to be played in order throughout. This was used to imitate the 52-17 work cycle (Thompson, 2014) without feeling under a timer to complete work. A CD ending would indicate a natural opportunity for a 15-20 minute break; this kept up productivity while reducing the pressure of deadlines.

Another methodology used was taken from Scrum; any features that needed to be implemented would be given a size of 1, 2, 4, or 8 hours in terms of how long it would take for them to be implemented. Anything larger than an 8 was split down into smaller tasks, for example the Level Editor was over 8, and was thus split into the UI for object selection (4), the Tile Selection System (4), and more.

# Development Process

Genre

Included in the client pitch was the need to justify the genre choice for the game. Bears is a puzzle game because of the opportunity it provides for expansion, and flexibility. As each level is self-contained; Bears can have expansions released with minimal effort or coding required, including an eventual way for players to share and download user-made levels in much the same way as multiple games on Steam can through the Steam Workshop (Valve Corp. 2014).

As for flexibility puzzle games such as Bears tend to have such simple mechanics, minimal controls, and bite-sized levels, that it would translate well to a medium such as mobile and tablet.

Level Creation

The first code written for Bears was a method of creating levels by typing into a text file, characters corresponding to objects. This was instrumental to the game’s development, as not only did it cut down on time that needed to be spent creating the levels, but it also became central to the modular approach used. Code was written so that during development, any level could be loaded by typing the name of the file into the unity editor, and having it instantiate out of prefabs at run-time.  
It is from this concept that the level editor was eventually created.

Features Cut

During the development process a number of features were considered for the game, while most of them were implemented in one way or another, a few were scrapped either due to being superfluous. Some of the features cut were

* The Inventor needing to collect an object to advance
* A dynamic camera
* Cut scenes between each level

The Inventor needing to collect an object to advance seemed as if it would interrupt the flow of gameplay for little gain, while the dynamic camera and the cut scenes proved to be unnecessary features in a top down puzzle game.

Features Included

One of the first things done once the initial concept of Bears was decided, was to create a Minimum Viable Product using only the essential features of the game to ensure the game could still be fun without extra features (Extra Credits, 2015). The features used for this were as follows:

* Two characters moving in tandem
* Walls
* Exits
* Bears

Once this was created it was clear that it was somewhat engaging, but would benefit from additional content, which could be added afterwards to make the main game more fun.  
The additional features used once the Minimum Viable Product was completed were as follows:

* Water hazards
* Pitfall hazards
* Buttons and Boxes

It was important to make sure each of these features could be used, and be fun, in more than one way in a level. An example of this is the pitfall hazards being able to be used to prevent the play from backtracking, but also being able to be used to trick bears into falling down the holes.

Testing

The game was tested in a step-by-step fashion, with all the bugs or unsatisfying gameplay moments being written down in note form for later review.

Different elements of the game were tested in different ways, below is an example of how a level is tested, followed by how the pitfall feature was tested.

Upon creating a new level, the shortest apparent solution to the level would be run through 15 times. If it was achieving its purpose (i.e. Level 2 is a tutorial level teaching the player the two different characters don’t have to make the same movements) and was fair for that stage in the game, then testing would proceed to the next step.  
The level would be played in as many ways as could reasonably be thought of, each time seeing if any bugs occur and also recording average step counts for different level solutions. Once this was finished, the gold silver and bronze medal step counts would be written down to be implemented in the game

The pitfall has three functions: it can be stepped over while it’s just a crack, it will become a pit when it is walked over as a crack, and it can be fallen down once it’s a pit. In order to properly test these two functions, all moving entities were moved over the pitfall as a crack, and then as a pit. Timing was also tested; to see from how many tiles away the player can bait a bear into a pit, and whether or not the pitfalls slow down any entities moving over them.  
Once it was seen that the bears rotated when moving over the pitfalls, the pitfalls were inspected until the reason was discovered. Once the problem with the collider was fixed the testing began again, it was determined that there were no further problems with the prefab and testing ended.

Structure

The levels were drafted out with the first five being tutorials in some way; each introducing a new concept or element to the game.  
For example, the first two levels only require the player walking down left and right, and pushing a box onto a button. The movement shows the player their goal will be downwards. The box and the button are placed directly next to the exit – which is sealed until the box is on the button, and the box cannot be manipulated at all other than to be pushed onto the button. This provides a situation in which the player discovers how exits, boxes, and buttons all interact without being explicitly told anything. The levels also end on the scoreboard, which shows the player their steps are being recorded and they get ranked based on this, implicitly telling them this game will involve more than just getting to the exits, but strategy in getting to the exits in as few steps as possible.

# Implementation

Required Game Aspects

The brief from the client included several aspects they needed to be included within the game, including

* Narrative Structure
* Gameplay with specific challenges
* Consideration of user interaction
* Enemies (or equivalent obstacles)
* GUI
* Sound Effects

These were all implemented throughout the course of the game development. Some examples below include how the narrative, the gameplay challenges, and the enemies were all developed and implemented.

Narrative

The narrative of the game was originally presented in a text window before the game began. However, this proved to be very unengaging for the player, and also very jarring for the minimal style of the game. The solution to this problem was to instead have the story represented in a cut scene shown before the game begins made up of pictures with short captions, in order to keep the player entertained whilst learning the story.

Challenges

The specific challenges the game presents are challenges of skill in both simple logic problem solving, and in lateral thinking. The way the game accomplishes this is through its medal system; when a level is completed the player receives a Gold/Silver/Bronze rank based on how many steps their characters took that level. This introduces a new gameplay element of completing the level in as few steps as possible, rather than just getting to the exit.

Interaction

How the player interacts with the environment – pushing boxes onto buttons, evading bears, using walls to line up the characters, etc. – is very central to the games main mechanic of having to control two characters at once. For many interactions the player could make in the game research went into how other games had achieved similar things in their puzzles, and how Bears could deviate and build upon that. An instance of this is for the boxes; inspiration was taken from Adventures of Lolo for the NES, in which boxes that block objects are used as obstacles to be pushed out of the way of the player’s path. (NESguide.com, 2007) (Refenes, 2015)

Enemies

There is only one real enemy in the game: the bear. The bear is introduced in the story as ambushing the player character, and separating him from his robot. Within the actual gameplay, the bears function is as more of an obstacle than an enemy with artificial intelligence; if the inventor is within the bear’s vision in any cardinal direction, the bear will charge until hitting something in its path. Originally the bears were much faster than in the final build, and disappeared upon collision with a wall. This was changed to keep a degree of challenge within the game.

GUI

Originally all level selection was going to be made within the confines of gameplay, i.e. having the player move the inventor around a pseudo level, having him walk onto specific tiles in order to select specific levels, level editor, quit etc. However it soon became apparent that not only would this be far too resource heavy, but that it would be very inefficient and frustrating for the player as well, as they would have to wait for the character to walk to the button they wanted to select rather than just being able to click it.

# Conclusion

Recommendation for Future Development

In future development, it is planned for the full Bears game to be completed, optimized, and ported to android devices via the Google Play Store. This will take full advantage of its bite-sized gameplay, and its simple control scheme, and its casual nature. Further upon this, depending on the scope of the game, a global leader board could be added, basing the scores off the best step counts of each user.

Reflection of development experience

A large key to the general success of Bears’ development was the realistic nature of the game’s scope. Early on in development it was decided that it would be better to develop a small polished demo for a game, which is a fun experience to play, rather than a large buggy demo, which would serve only to frustrate its players.  
Overall, this was achieved; the level editor – which was the most difficult feature to implement – is now fully functioning, and was worth the extra effort in order to provide players with something that makes the game more replayable.

The development of Bears has been a valuable learning experience, and has shown that it is extremely important to plan each detail of a game on paper from the very beginning of development, as this helps to visualise the end product and any problems or things to improve that may arise.

# Bibliography

Extra Credits, 2015. *Extra Credits – Making Your First Game: Minimum Viable Product – How To Scope Small and Start Right*. [video online] Available at: <https://www.youtube.com/watch?v=UvCri1tqIxQ> [Accessed Jan 28 2015]

NESguide.com, 2007. *Adventures of Lolo – NES Gameplay.* [video online] Available at: <https://www.youtube.com/watch?v=gC344LoZYdQ> [Accesed Feb 21 2015]

Thompson, D., 2014. A Formula for Perfect Productivity: Work for 52 Minutes, Break for 17. *The Atlantic,* [online] 17 September. Available at <http://www.theatlantic.com/business/archive/2014/09/science-tells-you-how-many-minutes-should-you-take-a-break-for-work-17/380369/> [Accessed 6 February 2015]

Refenes, T., 2015. Speaking the language: The Adventures of Lolo. *Yzc2NjQ1MzQyNDU2MjMyMz*, [blog] 21 Feb 2015. Available at <http://tommyrefenes.tumblr.com/search/lolo> [Accessed 21 Feb 2015]

Valve Corporation, 2014. *Octodad: Dadliest Catch.* [online] Available at <http://steamcommunity.com/workshop/browse/?appid=224480> [Accessed 4 May 2015]